LOW-RANK DATA MODELING VIA THE MINIMUM DESCRIPTION LENGTH PRINCIPLE

Ignacio Ramírez and Guillermo Sapiro

Department of Electrical and Computer Engineering, University of Minnesota

ABSTRACT

Robust low-rank matrix estimation is a topic of increasing interest, with promising applications in a variety of fields, from computer vision to data mining and recommender systems. Recent theoretical results establish the ability of such data models to recover the true underlying low-rank matrix when a large portion of the measured matrix is either missing or arbitrarily corrupted. However, if low rank is not a hypothesis about the true nature of the data, but a device for extracting regularity from it, no current guidelines exist for choosing the rank of the estimated matrix. In this work we address this problem by means of the Minimum Description Length (MDL) principle – a well established information-theoretic approach to statistical inference – as a guideline for selecting a model for the data at hand. We demonstrate the practical usefulness of our formal approach with results for complex background extraction in video sequences.

Index Terms— Low-rank matrix estimation, PCA, Robust PCA, MDL.

1. INTRODUCTION

The key to success in signal processing applications often depends on incorporating the right prior information about the data into the processing algorithms. In matrix estimation, low-rank is an all-time popular choice, with analysis tools such as Principal Component Analysis (PCA) dominating the field. However, PCA estimation is known to be non-robust, and developing robust alternatives is an active research field (see [1] for a review on low-rank matrix estimation). In this work, we focus on a recent robust variant of PCA, coined RPCA [1], which assumes that the difference between the observed matrix $\mathbf{Y}$, and the true underlying data $\mathbf{X}$, is a sparse matrix $\mathbf{E}$ whose non-zero entries are arbitrarily valued. It has been shown in [1] that $\mathbf{X}$ (alternatively, $\mathbf{E}$) can be recovered exactly by means of a convex optimization problem involving the rank of $\mathbf{Y}$ and the ℓ_1 norm of $\mathbf{E}$. The power of this approach has been recently demonstrated in a variety of applications, mainly computer vision (see [2] and http://perception.csl.uiuc.edu/matrix-rank/applications.html for examples).

However, when used as a pure data modeling tool, with no assumed "true" underlying signal, the rank of $\mathbf{X}$ in a PCA/RPCA decomposition is a parameter to be tuned in order to achieve some desired goal. A typical case is *model selection* [3, Chapter 7], where one wants to select the *size* of the model (in this case, rank of the approximation) in order to strike an optimal balance between the ability of the estimated model to generalize to new samples, and its ability to adapt itself to the observed data (the classic overfitting/underfitting trade-off in statistics). The main issue in model selection is how to formulate this balance as a cost function.

In this work, we address this issue via the Minimum Description Length (MDL) principle [4, 5].[1] MDL is a general methodology for assessing the ability of statistical models to capture regularity from data. The MDL principle can be regarded as a practical implementation of the Occam's razor principle, which states that, given two descriptions for a given phenomenon, the shorter one is usually the best. In a nutshell, MDL equates "ability to capture regularity" with "ability to compress" the data, using *codelength* or *compressibility* as the metric for measuring candidate models.

The resulting framework provides a robust, parameter-free low-rank matrix selection algorithm, capable of capturing relevant low-rank information in the data, as in the video sequences from surveillance cameras in the illustrative application here reported. From a theoretical standpoint, this brings a new, information theoretical perspective into the problem of low-rank matrix completion. Another important feature of an MDL-based framework such as the one here presented is that new prior information can be naturally and easily incorporated into the problem, and its effect can be assessed *objectively* in terms of the different codelengths obtained.

2. LOW-RANK MATRIX ESTIMATION/APPROXIMATION

Under the low-rank assumption, a matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$ can be written as $\mathbf{Y} = \mathbf{X} + \mathbf{E}$, where $\mathrm{rank}(\mathbf{X}) \ll \min\{m,n\}$ and $\|\mathbf{E}\| \ll \|\mathbf{Y}\|$, where $\|\cdot\|$ is some matrix norm. Classic PCA provides the best rank-k approximation to $\mathbf{Y}$ under the assumption that $\mathbf{E}$ is a random matrix with zero-mean IID Gaussian entries,

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\|_2\,, \quad \text{s.t.} \quad \mathrm{rank}(\mathbf{W}) \leq k. \tag{1}$$

However, PCA is known to be non-robust, meaning that the estimate $\hat{\mathbf{X}}$ can vary significantly if only a few coefficients in $\mathbf{E}$ are modified. This work, providing an example of introducing the MDL framework in this type of problems, focuses on a robust variant of PCA, RPCA, introduced in [1]. RPCA estimates $\mathbf{X}$ via the following convex optimization problem,

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}\|_1 + \lambda\,\|\mathbf{W}\|_*\,, \tag{2}$$

where $\|\mathbf{W}\|_* := \sum_i \sigma(\mathbf{W})_i$ is the nuclear norm of $\mathbf{W}$ ($\sigma(\mathbf{W})_i$ denotes the i-th singular value of $\mathbf{W}$). The rationale behind (2) is as follows. First, the ℓ_1 fitting term allows for large errors to occur in the approximation. In this sense, it is a robust alternative to the ℓ_2 norm used in PCA. The second term, $\lambda\,\|\mathbf{W}\|_*$, is a convex approximation to the PCA constraint $\mathrm{rank}(\mathbf{W}) \leq k$, merged into the cost function via a Lagrange multiplier λ.

This formulation has been recently shown to be notoriously robust, in the sense that, if a true low-rank matrix $\mathbf{X}$ exists, it can

Work supported by NSF, NGA, ONR, DARPA, ARO, and NSSEFF.

[1] While here we address the matrix formulation, the developed framework is applicable in general, including to sparse models, and such general formulation will be reported in our extended version of this work.

be recovered using (2) even when a significant amount of coefficients in $\mathbf{E}$ are arbitrarily large [1]. This can be achieved by setting $\lambda = 1/\sqrt{\max\{m,n\}}$, so that the procedure is parameter-free.

2.1. Low-rank approximation as dimensionality reduction

In many applications, the goal of low-rank approximation is not to find a "true" underlying matrix $\mathbf{X}$, but to perform what is known as "dimensionality reduction," that is, to obtain a succinct representation of $\mathbf{Y}$ in a lower dimensional subspace. A typical example is feature selection for classification. In such cases, $\mathbf{E}$ is not necessarily a small measurement perturbation, but a *systematic*, possibly large, error derived from the approximation process itself. Thus, RPCA arises as an appealing alternative for low-rank approximation.

However, in the absence of a true underlying signal $\mathbf{X}$ (and deviation $\mathbf{E}$), it is not clear how to choose a value of λ that produces a good approximation of the given data $\mathbf{Y}$ for a given application. A typical approach would involve some cross-validation step to select λ to maximize the final results of the application (for example, minimize the error rate in a classification problem).

The issue with cross-validation in this situation is that the best model is selected *indirectly* in terms of the final results, which can depend in unexpected ways on later stages in the data processing chain of the application (for example, on some post-processing of the extracted features). Instead, we propose to select the best low-rank approximation by means of a *direct measure* on the intrinsic ability of the resulting model to capture the desired regularity from the data, this also providing a better understanding of the actual structure of the data. To this end, we use the MDL principle, a general information-theoretic framework for model selection which provides means to define such a direct measure.

3. MDL-BASED LOW-RANK MODEL SELECTION

Consider a family $\mathcal{M}$ of candidate models which can be used to describe a matrix $\mathbf{Y}$ *exactly* (that is, losslessly) using some encoding procedure. Denote by $L(\mathbf{Y}|M)$ the description length, in bits, of $\mathbf{Y}$ under the description provided by a given model $M \in \mathcal{M}$. MDL will then select the model $\hat{M} \in \mathcal{M}$ for $\mathbf{Y}$ for which $L(\mathbf{Y}|\hat{M})$ is minimal, that is $\hat{M} = \arg\min_{M\in\mathcal{M}} L(\mathbf{Y}|M)$. It is a standard practice in MDL to use the *ideal* Shannon code for translating probabilities into codelengths. Under this scheme, a sample value y with probability $P(y)$ is assigned a code with length $L(y) = -\log P(y)$ (all logarithms are taken on base 2). This is called an ideal code because it only specifies a codelength, not a specific binary code, and because the codelengths produced can be fractional.

By means of the Shannon code assignment, encoding schemes $L(\cdot)$ can be defined naturally in terms of probability models $P(\cdot)$. Therefore, the art of applying MDL lies in defining appropriate probability assignments $P(\cdot)$, that exploit as much prior information as possible about the data at hand, in order to maximize compressibility. In our case, there are two main components to exploit. One is the low-rank nature of the approximation $\mathbf{X}$, and the other is that most of the entries in $\mathbf{E}$ will be small, or even zero (in which case $\mathbf{E}$ will be *sparse*). Given a low-rank approximation $\mathbf{X}$ of $\mathbf{Y}$, we describe $\mathbf{Y}$ as the pair $(\mathbf{X},\mathbf{E})$, with $\mathbf{E} = \mathbf{Y} - \mathbf{X}$. Thus, our family of models is given by $\mathcal{M} = \{(\mathbf{X},\mathbf{E}) : \mathbf{Y} = \mathbf{X} + \mathbf{E}, \operatorname{rank}(\mathbf{X}) \leq \operatorname{rank}(\mathbf{Y})\}$. As $\mathbf{E} = \mathbf{Y} - \mathbf{X}$, we index $\mathcal{M}$ solely by $\mathbf{X}$. With these definitions, the description codelength of $\mathbf{Y}$ is given by $L(\mathbf{Y}|\mathbf{X}) := L(\mathbf{X}) + L(\mathbf{E})$. Now, to exploit the low rank of $\mathbf{X}$, we describe it in terms of its reduced SVD decomposition,

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^{\mathsf{T}}\ \mathbf{U} \in \mathbb{R}^{m\times k},\ \Sigma \in \mathbb{R}^{k\times k},\ \mathbf{V} \in \mathbb{R}^{k\times n}, \qquad (3)$$

where k is the rank of $\mathbf{X}$ (the zero-eigenvalues and the respective left and right eigenvectors are discarded in this description). We now have $L(\mathbf{X}) = L(\mathbf{U}) + L(\Sigma) + L(\mathbf{V})$. Clearly, such description will be short if $\operatorname{rank}(\mathbf{X})$ is significantly smaller than $\max\{m,n\}$. We may also be able to exploit further structure in $\mathbf{U}$, Σ and $\mathbf{V}$.

3.1. Encoding Σ

The diagonal of Σ is a non-increasing sequence of k positive values. However, no safe assumption can be made about the magnitude of such values. For this scenario we propose to use the *universal prior for integers*, a general scheme for encoding arbitrary positive integers in an efficient way [6],

$$L(j) = \log^* j := \log j + \log\log j + \ldots + \log 2.865, \qquad (4)$$

where the sum stops at the first non-positive summand, and $\log 2.865$ is added to satisfy Kraft's inequality (a requirement for the code to be uniquely decodable, see [7, Chapter 5]). In order to apply (4), the diagonal of Σ, $\operatorname{diag}(\Sigma)$, is mapped to an integer sequence via $[10^{16}\operatorname{diag}(\Sigma)]$, where $[\cdot]$ denotes rounding to nearest integer (this is equivalent to quantizing $\operatorname{diag}(\Sigma)$ with precision $\delta_\Sigma = 10^{-16}$).

3.2. Encoding U and V, general case

By virtue of the SVD algorithm, the columns of $\mathbf{U}$ and $\mathbf{V}$ have unit norm. Therefore, the most general assumption we can make about $\mathbf{U}$ and $\mathbf{V}$ is that their columns lie on the respective m-dimensional and n-dimensional unit spheres.

An efficient code for this case can be obtained by encoding each column of $\mathbf{U}$ and $\mathbf{V}$ in the following manner. Let $\mathbf{u}_i$ be a column of $\mathbf{U}$ ($\mathbf{V}$ is similarly encoded). Since $\mathbf{u}_i$ is assumed to be distributed uniformly over the m-dimensional unit sphere, the marginal cumulative density function of the first element u_{1i}, $F(u_{1i}) = P(x \leq u_{1i})$, corresponds to the proportion of vectors $\mathbf{u}$ that lie on the *unit spherical cap* of height $h = 1 + u_{1i}$ (see Figure 1(a)). This proportion is given by $F(u_{1i}) = A_m(1+u_{1i},1)/S_m(1)$ where $A_m(h,r)$ and $S_m(r)$ are the area of spherical cap of height h and the total surface area of the m-dimensional sphere of radius r respectively. These are given for the case $0 \leq h \leq r$ ($-1 \leq u_{1i} \leq 0$) by (see [8]),

$$\begin{aligned} A_m(h,r) &= \frac{1}{2}S_m(r)I((2hr-h^2)/r^2\,;\,\frac{m-1}{2},\frac{1}{2}) \\ S_m(r) &= 2\pi^{m/2}r^{m-1}\Gamma^{-1}(m/2), \end{aligned}$$

where $I(x\,;a,b) = \frac{\int_0^x t^{a-1}(1-t)^{b-1}dt}{B(a,b)}$, and $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt$ are the regularized incomplete Beta function and the Beta function of parameters a,b respectively, and $\Gamma(\cdot)$ is the Gamma function. When $r < h \leq 2r$ we simply have $A_m(h,r) = 1 - A_m(2r-h,r)$. For encoding u_{1i} we have $r = 1$ so that

$$F(u_{1i}) = (1/2)I(1-u_{1i}^2;(m-1)/2,1/2), -1 \leq u_{1i} \leq 0, \quad (5)$$

since $2h - h^2 = h(2-h) = (1+u_{1i})[2-(1+u_{1i})] = 1 - u_{1i}^2$. Finally, we compute the Shannon codelength for u_{1i} as

$$\begin{aligned} p(u_{1i}) = F'(u_{1i}) &\overset{(a)}{=} \frac{(1-u_{1i}^2)^{(m-3)/2}(u_{1i}^2)^{-1/2}}{2\cdot B\left(\frac{m-1}{2},\frac{1}{2}\right)}(-2u_{1i}) \\ &= -\operatorname{sgn}(u_{1i})(1-u_{1i}^2)^{(m-3)/2}[B((m-1)/2,1/2)]^{-1} \\ -\log p(u_{1i}) = -\frac{m-3}{2}&\log(1-u_{1i}^2) + \log B((m-1)/2,1/2), \end{aligned}$$

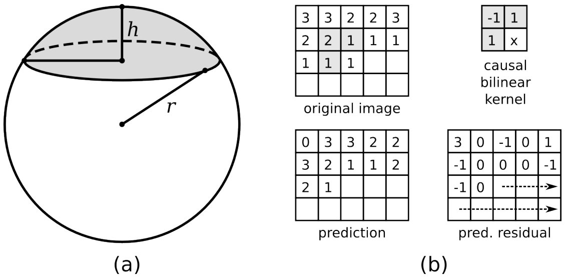

Fig. 1. (a) The spherical cap of radius r and height h (shown in gray). (b) Causal bilinear prediction of smooth 2D images.

where in (a) we applied the Fundamental Theorem of Calculus to the definition of $F(h)$ and the chain rule for derivatives.

With u_{1i} encoded, the vector $(u_{2i}, u_{3i}, \ldots, u_{mi})$ is uniformly distributed on the surface of the $(m-1)$-dimensional sphere of radius $r' = 1 - |u_{1i}|$, and we can apply the same formula to compute the probability of u_{2i}, $F(u_{2i}) = A_{m-1}(u_{2i} + r', r')/S_{m-1}(r')$.

Finally, to encode the next column $\mathbf{u}_{i+1}$, we can exploit its orthogonality with respect to the previous ones and encode it as a vector distributed uniformly over the $m - i$ dimensional sphere corresponding to the intersection of the unit sphere and the subspace perpendicular to $[\mathbf{u}_1, \ldots, \mathbf{u}_i]$.

In order to produce finite descriptions $L(\mathbf{U})$ and $L(\mathbf{V})$, both $\mathbf{U}$ and $\mathbf{V}$ also need to be quantized. We choose the quantization steps for $\mathbf{U}$ and $\mathbf{V}$ adaptively, using as a starting point the empirical standard deviation of a normalized vector, that is, $\delta_u = \sqrt{1/m}$ and $\delta_v = \sqrt{1/n}$ respectively, and halving these values until no further decrease in the overall codelength $L(\mathbf{Y}|\mathbf{X})$ is observed.

3.3. Encoding U predictively

If more prior information about $\mathbf{U}$ and $\mathbf{V}$ is available, it should be used as well. For example, in the case of our example application, the columns of $\mathbf{Y}$ are consecutive frames of a video surveillance camera. In this case, the columns of $\mathbf{U}$ represent "eigen-frames" of the video sequence, while $\mathbf{V}$ contains information about the evolution in time of those frames (this is clearly observed in figures 2 and 3). Therefore, the columns of $\mathbf{U}$ can be assumed to be piecewise smooth, just as normal static images are. To exploit this smoothness, we apply a predictive coding to the columns of $\mathbf{U}$. Concretely, to encode the i-th column $\mathbf{u}_i$ of $\mathbf{U}$, we reshape it as an image $\mathbf{B}$ of the same size as the original frames in $\mathbf{Y}$. We then apply a causal bilinear predictor to produce an estimate of $\mathbf{B}$, $\hat{\mathbf{B}} = \{\hat{b}_{jl}\}$ where $\hat{b}_{jl} = b_{jl} - b_{j(l-1)} - b_{(j-1)l} + b_{(j-1)(l-1)}$, assuming out-of-range pixels to be 0. The prediction residual $\tilde{\mathbf{B}} = \mathbf{B} - \hat{\mathbf{B}}$ is then encoded in raster scan as a sequence of Laplacian random variables with unknown parameter θ_u^i. This encoding procedure, common in predictive coding, is depicted in Figure 1(b).

Since the parameters $\{\theta_u^i, i = 1, \ldots, k\}$ are unknown, we need to encode them as well to produce a complete description of $\mathbf{Y}$. In MDL, this is done using the so-called *universal encoding schemes*, which can be regarded as a generalization of classical Shannon encoding to the case of distributions with unknown parameters (see [5] for a review on the subject). In this work we adopt the so-called *universal two-part codes*, and apply it to encode each column $\mathbf{u}_i$ separately. Under this scheme, the unknown Laplacian parameter for θ_u^i is estimated via Maximum Likelihood, $\hat{\theta}_u^i(\mathbf{u}_i)$, and quantized with precision $1/\sqrt{m}$, thus requiring $L(\hat{\theta}_u^i) = \frac{1}{2}\log m + c_1$ bits. Given the quantized $\hat{\theta}_u^i$, $\mathbf{u}_i$ is described using the discretized Laplacian distribution $L(\mathbf{u}_i) = -\log P(\mathbf{u}_i|\hat{\theta}_u^i(\mathbf{u}_i)) + c_2$. Here c_1 and c_2 are constants which can be disregarded for optimization purposes. It was shown in [4] that the precision $1/\sqrt{m}$ asymptotically yields the shortest two-parts codelength.

3.4. Encoding V predictively

We also expect a significant redundancy in the time dimension, so that the columns of $\mathbf{V}$ are also smooth functions of time (in this case, sample index $j = 1, 2, \ldots, n$). In this case, we apply a first order causal predictive model to the columns of $\mathbf{V}$, by encoding them as sequences of prediction residuals, $\tilde{\mathbf{v}}_i = (\tilde{v}_{i1}, \tilde{v}_{i2}, \ldots, \tilde{v}_{in})$, with $\tilde{v}_{ij} = v_{ij} - v_{i(j-1)}$ for $j > 1$ and $\tilde{v}_{i1} = v_{i1}$. Each predicted column $\mathbf{v}_i$ is encoded as a sequence of Laplacian random variables with unknown parameter θ_v^i. As with $\mathbf{U}$, we use a two-parts code here to describe the data and the unknown Laplacian parameters together. This time, since the length of the columns is n, the codelength associated to each θ_v^i is $L(\theta_v^i) = \frac{1}{2}\log n$.

3.5. Encoding E

We exploit the (potential) sparsity of $\mathbf{E}$ by first describing the indexes of its non-zero locations using an efficient universal two-parts code for Bernoulli sequences known as Enumerative Code [9], and then the non-zero values at those locations using a Laplacian model. In the specific case of the experiments of Section 4, we encode each row of $\mathbf{E}$ separately. Because each row of $\mathbf{E}$ corresponds to the pixel values at a fixed location across different frames, we expect some of these locations to be better predicted than others (for example, locations which are not occluded by people during the sequences), so that the variance of the error (hence the Laplacian parameter) will vary significantly from row to row. As before, the unknown parameters here are dealt with using a two-parts coding scheme.

3.6. Model selection algorithm

To obtain the family of models $\mathcal{M}$ corresponding to all possible low-rank approximations of $\mathbf{Y}$, we apply the RPCA decomposition (2) for a decreasing sequence of values of λ, $\{\lambda_t : t = 1, 2, \ldots\}$ obtaining a corresponding sequence of decompositions $\{(\mathbf{X}_t, \mathbf{E}_t), t = 1, 2, \ldots\}$. We obtain such sequence efficiently by solving (2) via a simple modification of the Augmented Lagrangian-based (ALM) algorithm proposed in [10] to allow for *warm restarts*, that is, where the initial ALM iterate for computing $(\mathbf{X}_t, \mathbf{E}_t)$ is $(\mathbf{X}_{t-1}, \mathbf{E}_{t-1})$. We then select the pair $(\mathbf{X}_{\hat{t}}, \mathbf{E}_{\hat{t}})$, $\hat{t} = \arg\min_t\{L(\mathbf{X}_t) + L(\mathbf{E}_t)\}$.

4. RESULTS AND CONCLUSION

In order to have a reference base, we repeated the experiments performed in [2] using our algorithm. These experiments consist of frames from surveillance cameras which look at a fixed point where people pass by. The idea is that, if frames are stacked as columns of $\mathbf{Y}$, the background can be well modeled as a low-rank component of $\mathbf{Y}$ ($\mathbf{X}$), while the people passing by appear as "spurious errors" ($\mathbf{E}$). Clearly, if the background in all frames is the same, it can be very well modeled as a rank-1 matrix where all the columns are equal. However, lighting changes, shadows, and reflections, "raise" the rank of the background, and the appropriate rank needed to model the background is no longer obvious.

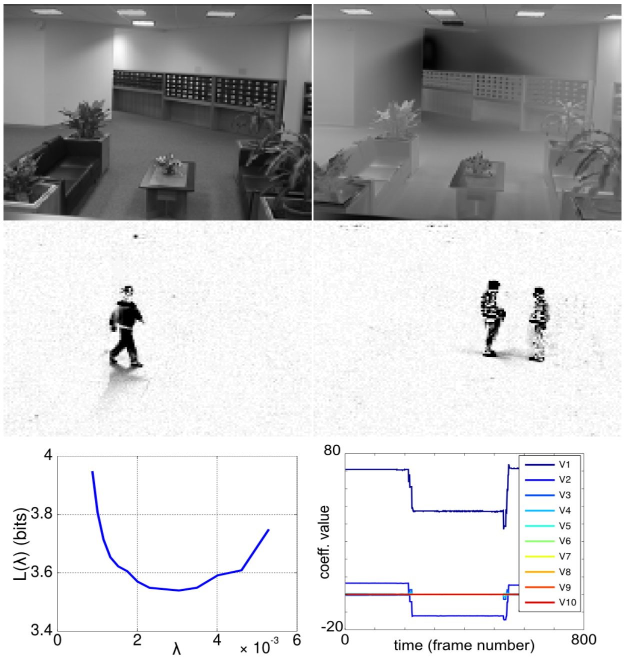

Fig. 2. Results for the "Lobby" sequence (see text for a description of the above pictures and graphs). The rank of the approximation decomposition for this case is $k = 10$. The moment where the lights are turned off is clearly seen here as the "square pulse" in the middle of the first two right-eigenvectors (bottom-right figure). Also note how $\mathbf{u}_2$ (top-right) compensates for changes in shadows.

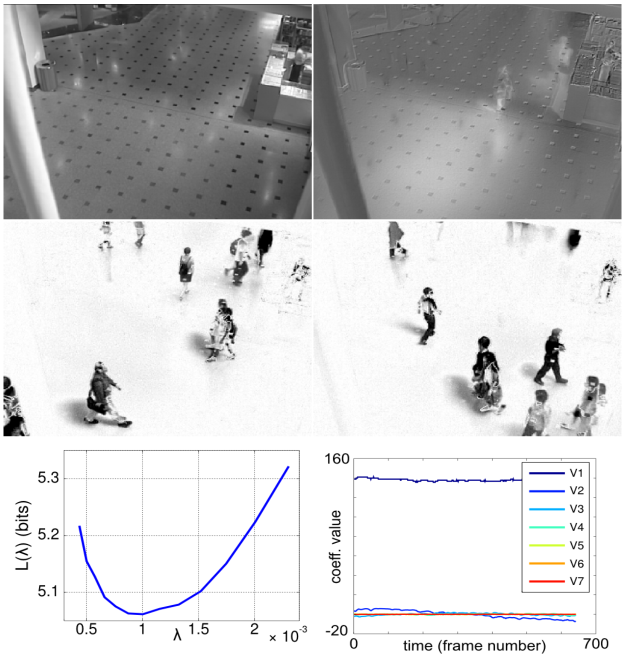

Fig. 3. Results for the "ShoppingMall" sequence (see text for a description of the above pictures and graphs). In this case, the rank of the approximation decomposition is $k = 7$. Here, the first left-eigenvector models the background, whereas the rest tend to capture people that stood still for a while. Here we see the "phantom" of two such persons in the second left-eigenvector (top-right).

Concretely, the experiments here described correspond to two sequences: "Lobby" and "ShoppingMall," whose corresponding results are summarized respectively in figures 2 and 3.[2] At the top of both figures, the first two left-eigenvectors $\mathbf{u}_1$ and $\mathbf{u}_2$ of $\mathbf{X}$ are shown as 2D images. The middle shows two sample frames of the error approximation. The L-vs-λ curve is shown at the bottom-left (note that the best λ is *not* the one dictated by the theory in [1], which are 0.007 for Lobby and 0.0035 for ShoppingMall, both outside of the plotted range), and the scaled right-eigenvectors $\sigma_i \mathbf{v}_i$ are shown on the bottom-right. In both cases, the resulting decomposition recovered the low-rank structure correctly, including the background, its changes in illumination, and the effect of shadows. It can be appreciated in the figures 2-3 how such approximations are naturally obtained as combinations of a few significant eigen-vectors, starting with the average background, followed by other details.

4.1. Conclusion

In summary, we have presented an MDL-based framework for low-rank data approximation, which combines state-of-the-art algorithms for robust low-rank decomposition with tools from information theory. This framework is able to capture the underlying low-rank information on the experiments that we performed, out of the box, and without any hand parameter tuning, thus constituting a promising competitive alternative for automatic data analysis and feature extraction.

5. REFERENCES

[1] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, May 2011.

[2] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization," in *Adv. NIPS*, Dec. 2009.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, 2nd edition, Feb. 2009.

[4] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[5] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. IT*, vol. 44, no. 6, pp. 2743–2760, 1998.

[6] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, Singapore: World Scientific, 1992.

[7] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley and Sons, Inc., 2 edition, 2006.

[8] S. Li, "Concise formulas for the area and volume of a hyperspherical cap," *Asian J. Math. Stat.*, vol. 4, pp. 66–70, 2011.

[9] T. M. Cover, "Enumerative source encoding," *IEEE Trans. IT*, vol. 19, no. 1, pp. 73–77, 1973.

[10] Z. Lin, M. Chen, and Y. Ma, "The Augmented Lagrange Multiplier Method for exact recovery of corrupted low-rank matrices," http://arxiv.org/abs/1009.5055.

[2] The full videos can be viewed at `http://www.tc.umn.edu/~nacho/lowrank/`.